Quantifying uncertainty with ensembles of surrogates for blackbox optimization

C. Audet, S. Le Digabel, R. Saltet

Quantifying uncertainty with ensembles of surrogates for blackbox optimization

Charles Audet

Sébastien Le Digabel

Renaud Saltet

GERAD & Département de Mathématiques et de Génie Industriel, Polytechnique Montréal, Montréal (Québec), Canada H3C 3A7

charles.audet@gerad.ca
sebastien.le.digabel@gerad.ca
renaud.saltet@polymtl.ca

Abstract : This work is in the context of blackbox optimization where the functions defining the problem are expensive to evaluate and where no derivatives are available. A tried and tested technique is to build surrogates of the objective and the constraints in order to conduct the optimization at a cheaper computational cost. This work proposes different uncertainty measures when using ensembles of surrogates. The resulting combination of an ensemble of surrogates with our measures behaves as a stochastic model and allows the use of efficient Bayesian optimization tools. The method is incorporated in the search step of the mesh adaptive direct search (MADS) algorithm to improve the exploration of the search space. Computational experiments are conducted on seven analytical problems, two multi-disciplinary optimization problems and two simulation problems. The results show that the proposed approach solves expensive simulation-based problems at a greater precision and with a lower computational effort than stochastic models.

Keywords: Blackbox optimization, Derivative-free optimization, Ensembles of surrogates, Mesh adaptive direct search, Bayesian optimization

1 Introduction

This work considers the constrained optimization problem

$$\min_{x \in \mathcal{X}} \quad f(x)$$
$$\text{s.t.} \quad x \in \Omega \qquad (P)$$

where $\mathcal{X}$ is a subset of $\mathbb{R}^n$; Ω denotes the feasible set $\big\{x \in \mathcal{X} \mid c_j(x) \le 0, \; j \in \{1, 2, \ldots, m\}\big\}$, where the functions $c_j : \mathbb{R}^n \to \overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ are the constraint functions of the problem; and $f : \mathbb{R}^n \to \overline{\mathbb{R}}$ is the objective function. The set $\mathcal{X}$ contains the points that satisfy *unrelaxable* constraints [39]: every point explored during the optimization process must lie in $\mathcal{X}$ either because f is not defined elsewhere or because a point outside $\mathcal{X}$ has no meaning in the original problem, e.g., a negative length or a probability greater than one. The set $\mathcal{X}$ typically represents bound constraints of the form $\mathcal{X} = \{x \in \mathbb{R}^n \mid \ell \le x \le u\}$ where ℓ and u are vectors of $\{\mathbb{R} \cup \{-\infty\}\}^n$ and $\{\mathbb{R} \cup \{+\infty\}\}^n$, respectively. The functions c_j, $j \in \{1, 2, \ldots, m\}$, denote *relaxable* constraints, which means that they can be violated during the optimization process, however, the final solution must satisfy these constraints.

In *blackbox optimization* (BBO), no information is available on the functions f and c_j, $j \in \{1, 2, \ldots, m\}$, beyond the mere values they produce, hence the *blackbox* designation. In particular, no derivatives can be used either because they are especially hard to estimate or because they do not exist. Designing algorithms that do not use derivatives is referred to as *Derivative-Free Optimization* (DFO). This context typically occurs when the functions are the results of numerical simulations. Consequently, a blackbox is assumed to be costly, i.e., one evaluation might take seconds [20], minutes [5, 47], hours [15, 65] or even days [41]. BBO consists in designing algorithms capable of finding the best possible solution to such a problem with a given budget of function evaluations. For a better understanding of the theoretical importance of the existence of derivatives in DFO, see [12]. Two reference books are available in DFO and BBO [11, 23] as well as an extended review [38].

BBO algorithms can be roughly divided in two categories: *direct-search* methods and methods using *surrogates*. Direct-search algorithms only use comparisons between points and no other information like an approximation of derivatives. This philosophy has led to fruitful algorithmic frameworks such as the state-of-the-art algorithms *generalized pattern search* (GPS) [64] and *mesh-adaptive direct search* (MADS) [8], the latter is described in this article. Both frameworks lie on the search-poll paradigm [4]: the *search* step offers flexibility for the user to implement any method they see fit for the problem to optimize, while the *poll* step imposes more rigid procedures in order to guaranty convergence and explore further the surroundings of the best known solution. The second category resorts to *surrogates*, i.e., functions that are expected to mimic the behaviour of the objective and the constraints while being significantly cheaper to evaluated. A surrogate can be a simplified and static version of the blackbox that do not evolve over the optimization, or a dynamic surrogate that is based on regression or interpolation on the previously evaluated points. A dynamic surrogate is also called a model. One can then expect that minimizing a surrogate of the objective while satisfying surrogates of the constraints will lead to a promising new candidate point for the true problem. The combination of direct-search and model-based methods has proven fruitful. Surrogates can be incorporated either in a subproblem embedded in the search step to find candidates points, or as means of ranking candidate points of the poll step when opportunistic strategies are used.

Among the model-based methods, ensembles of models and stochastic models are two efficient techniques. Ensembles of models consist in giving each model a weight that is supposed to reflect its quality, and the combination of all weighted models yields an *aggregate model* that can be used as a standard surrogate. Stochastic models not only produce a prediction at a given point but also a measure of the uncertainty on this prediction, which is fit for Bayesian optimization. Ensembles of models and stochastic models have both proven efficient but remain fundamentally separated. Using several deterministic surrogates naturally produces a deterministic aggregate model which is incompatible with Bayesian optimization. If some of the models used are stochastic though, the provided uncertainty can

be exploited by all the models as in [69]. However, typical stochastic surrogates as Gaussian processes (GPs) become particularly costly to train as the training set grows. The proposed approach is to identify areas where the predictions of the models differ from each other in order to derive some form of uncertainty. The idea of using the correlation between several models to guide the optimization has already been tackled in the literature. In [29], the deviation between the predicted values of several models is used *a posteriori* to check the overall quality of the aggregate model. In [45], two surrogates of the objective are available: a low-fidelity one that is cheap to compute but not accurate and a high-fidelity one that is in contrast more expensive and more reliable. Then RBF models of the two surrogates are computed and the correlation between them is used to choose which surrogate, high or low-fidelity, to evaluate next. In [63], the correlation between variable fidelity co-kriging models is exploited to determine which model to evaluate. In [55], the correlation between variable fidelity multi-level generalized co-kriging models is incorporated in an extended probability of improvement.

The contribution of this work is an extension to ensembles of models when used in the form of aggregate models. For a point x of the search space $\mathcal{X}$, the extended aggregate models produce not only a prediction $\hat{f}(x)$, but also an uncertainty $\hat{\sigma}(x)$, therefore imitating a stochastic model. The resulting surrogate is then exploited in the search step of MADS in subproblems inspired by Bayesian optimization. The proposed approach has been tested on seven analytical problems, two multi-disciplinary optimization problems and two simulation problems. It has been compared to other versions of MADS as well as two other BBO solvers. Results show that the proposed extended aggregate models manage to find solutions of most of the difficult real-world problems at a greater precision than the other algorithms, and with less computational effort than the competing stochastic models.

The manuscript is structured as follows. Section 2 introduces ensemble of surrogates and the Bayesian optimization framework as well as a high-level description of the MADS algorithm. Section 3 describes the quantification of uncertainty when using ensembles of models and our incorporation of the resulting extended aggregtae models into the MADS algorithm. Section 4 shows the computational results on the set of problems. A concluding discussion is proposed in Section 5.

2 Background

This section describes the use of surrogates in BBO, with a special focus on stochastic surrogates and ensembles of surrogates, as well as the MADS algorithm.

2.1 Surrogates in BBO

A common approach in BBO uses *surrogates* of the objective and the constraints in order to guide the optimization. A surrogate shares similarities with the true functions of the problem while being significantly cheaper. Two types of surrogates can be distinguished: *static surrogates* and *dynamic models*. A static surrogate is a simplified version of the blackbox that can be obtained for example through a simplified physics model, a coarser mesh in a finite elements simulation or a looser stopping criterion in a numerical method. Such a surrogate is fixed and does not evolve over the optimization, hence the *static* designation. On the other hand, a dynamic model is an interpolation or regression model that approaches the true functions by fitting previously evaluated sample points. Since it attempts to approximate the true function, the *model* designation is more appropriate than surrogate in this case. Common models used in BBO are polynomial response surfaces (PRS) [3, 48] and especially quadratic models [22, 24, 25], radial basis functions (RBF) [16, 36, 52, 53, 54, 62, 71, 72], support vector machines (SVM) [67], kernel smoothing (KS) [3], and Gaussian processes (GPs) [17, 50, 51, 56].

Surrogates are basically used for two purposes in BBO: finding new candidate points and ranking existing candidate points before evaluation by the true problem. The surrogate management framework [18] establishes the interplay between the surrogate evaluations and the true evaluations and is described in Algorithm 1 as in [11, Chapter 13].

Algorithm 1: Surrogate management framework.

1. ***Exploration using the surrogate***
   Use the surrogate problem to generate a list $\mathcal{L}$ of candidate points
   Evaluate the true functions at points in $\mathcal{L}$ in an opportunistic way
   If a new incumbent solution is found, go to 3; otherwise go to 2
2. ***Ranking using the surrogate***
   Use the optimization algorithm to generate a list $\mathcal{L}$ of candidate points
   Use the surrogate functions to order the points in $\mathcal{L}$
   Evaluate the true functions at points in $\mathcal{L}$ in an opportunistic way
3. ***Parameters update***
   Update algorithmic parameters
   Check stopping criteria or go to 4
4. ***Model update (optional)***
   Update the model by using the new values of the true functions obtained in 1 and 2

For extended reviews on model-based or model-assisted optimization, see [12, 70] and [38, Section 2.2].

2.1.1   Ensemble of models

In an optimization context, there is not one type of models that dominates the others [3, 29]. Even for a given problem, the best performance can be obtained with different models depending on the initial sampling [29]. A tempting strategy is to resort to several models simultaneously. This idea has proven efficient in several works [3, 13, 19, 29, 48, 68, 73] in which an ensemble of models is used to build an aggregate model defined by

$$\hat{f}(x) = \sum_{p=1}^{s} w^p \tilde{f}^p(x) \tag{1}$$

where $s \geq 1$ is the number of models; $\tilde{f}^1, \tilde{f}^2, \ldots, \tilde{f}^s$ are s models of the objective f built from sample points; and $w^1, w^2, \ldots, w^s$ are positive weights such that $\sum_{p=1}^{s} w^p = 1$. The main difficulty lies in the attribution of the weights that must reflect the quality of the models. To do so, weights can be attributed so that the error of the aggregate model will be minimal, thus introducing an optimization subproblem [3, 68]. Another approach is to compute an error metric $\mathcal{E}^p$ for each model $\tilde{f}^p$ and then attribute a weight w^p that is a function of $\mathcal{E}^p$ [13, 29, 48]. In this type of strategy, an error metric must be chosen first. Common metrics for this purpose are statistical measures with cross-validation like root mean square error (RSME) and predicted residual sum of squares (PRESS) that take into account the gaps between the values of the models and those of the true objective. In [13], the authors propose an error metric that is not statistical but is rather a measure of a model's capacity to rank points in the same order as the objective would do. The rationale for this latter metric is that in a BBO context a good model does not necessarily approximate well the values of the true function but is rather capable of discriminating candidate points and telling apart promising ones.

Once an error metric is chosen, weights must be attributed accordingly. For instance, in [29], the authors propose the three following options:

$$w^p \propto \mathcal{E}^{\text{tot}} - \mathcal{E}^p$$
$$\text{or } w^p \propto \mathbb{1}_{\mathcal{E}^p = \mathcal{E}^{\min}}$$
$$\text{or } w^p \propto (\mathcal{E}^p + \alpha \mathcal{E}^{\text{av}})^\beta$$

where $\mathcal{E}^{\text{tot}}$ is the total error of all models, $\mathcal{E}^{\min}$ is the minimal error, $\mathcal{E}^{\text{av}}$ is the average error, and $\alpha < 1$ and $\beta < 0$ are adjustable parameters.

2.1.2   Bayesian optimization

In general, models may be trusted in areas where the true functions have been sufficiently sampled. But the further away from these areas, the less accurate are the models. In Bayesian optimization, the

objective function is interpreted as a stochastic process: an *a priori* distribution $P[f]$ is assumed, then with the set of sample points $\mathbb{X}$ and a likelihood model $P[\mathbb{X} \mid f]$, an *a posteriori* distribution $P[f \mid \mathbb{X}]$ is built thanks to Bayes' rule. Consequently, for any point x of the search space, a stochastic model not only produces a prediction $\mu(x)$ but also a measure of uncertainty on that prediction $\sigma(x)$. If properly exploited, this uncertainty enables to explore areas in which the model confesses to be unreliable, instead of spending the entire budget on restricted areas. This is called the compromise between exploration and exploitation. Commonly used stochastic models are generalized linear models [49], Gaussian processes [51], and dynamic trees [59].

The compromise between exploration and exploitation is then realized with an acquisition function. A simple example is the upper confidence bound (UCB) [58] that takes into account the most optimistic value, i.e., minimizes $\mu(x) - \kappa\sigma(x)$. A more sophisticated instance is the probability of improvement (PI) [33] which is the probability that the objective decreases from the best known value at a given point. Finally, a very popular example is the expected improvement (EI) [43] that not only takes into account the probability of decrease but also the expected amplitude thereof.

Figure 1 shows an example of Gaussian process regression - also know as *kriging* - as well as the resulting expected improvement on a one-dimensional objective function. The dashed curve represents the objective $f : x \mapsto x \sin x$; the five dots are the sample points; the curve interpolating the dots is the prediction $\mu : x \mapsto \mu(x)$; an filled area represents the 95% confidence interval given at any point x by $[\mu(x) + 1.96\sigma(x), \ \mu(x) - 1.96\sigma(x)]$. The curve at the bottom represents - in a different scale - the expected improvement EI. The resulting candidate point maximizes EI and is indicated by the vertical dashed line. Maximizing EI is a method introduced by Jones et al. [34] that is efficient and easy
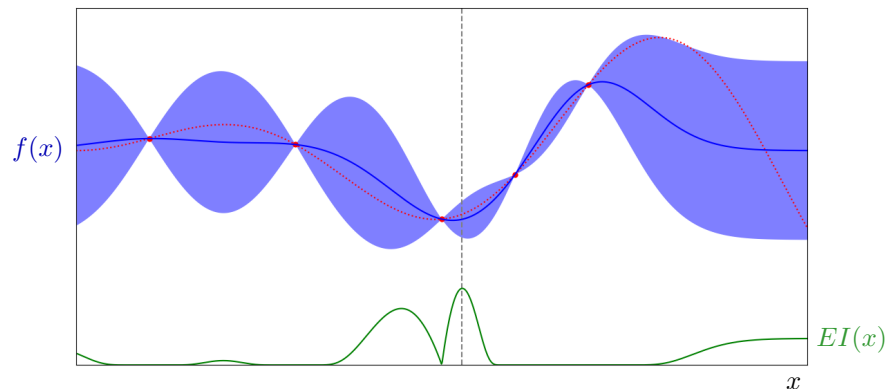
Figure 1: **Kriging and expected improvement (EI) on** $f : x \mapsto x \sin x$.

to grasp. More elaborated techniques have since been developed. Talgorn et al. [61] propose various formulations for the surrogate subproblem that use multiple acquisition functions at once.

For an extended literature review on Bayesian optimization, see [31, 57].

2.2 The MADS algorithm

MADS [8] is a direct-search algorithmic framework that follows a search-poll paradigm in which the mandatory poll step guaranties the convergence and the optional search step gives room for flexible exploration techniques. In order to ensure convergence, every candidate point must lie on a *mesh* defined at Iteration k by

$$M^k = \left\{ x + \delta^k Dy \ : \ x \in V^k, \ y \in \mathbb{N}^{n_D} \right\} \subset \mathbb{R}^n$$

where $V^k \subset \mathbb{R}^n$ is the *cache*, i.e., the set of all evaluated points up to Iteration k; $\delta^k > 0$ is the *mesh size parameter*; and D is a fixed matrix of $\mathbb{R}^{n \times n_D}$, the columns of which represent n_D directions of $\mathbb{R}^n$. Before the algorithm starts, V^0 is the set of one or more initial points provided by the user.

The search step enables to use various strategies to explore the space of variables. When the search is unsuccessful, i.e., when no better solution is found, a poll step is launched. Every candidate point generated during the poll step must lie within a frame centred around the incumbent solution x^k and which size is parameterized by the *poll size parameter* $\Delta^k \geq \delta^k$.

At the end of an iteration, the mesh and poll size parameters are updated depending on the outcome. If the iteration is unsuccessful both are increased, and conversely, if the iteration is successful both are decreased in such a way that the set of possible directions during the poll gets richer. In this work, OrthoMADS [2] is used to deterministically generate $2n$ orthogonal directions at the poll step. A high-level description of MADS is given Algorithm 2.

Algorithm 2: The Mesh Adaptive Direct-Search algorithm (MADS).

---

0. Initialization
  $V^0 \subset \mathbb{R}^n$: set of starting points
  $\Delta^0 \geq \delta^0 > 0$: initial mesh and poll size parameters
  $k \leftarrow 0$: iteration counter
1. Search (optional)
  Evaluate a finite set of points included in the mesh M^k.
  If the search if successful, go to 3, otherwise go to 2.
2. Poll
  Evaluate a finite set of points included in the poll frame.
3. Update parameters
  Update the cache V^{k+1} with the newly sampled points.
  Update the mesh and poll size parameters δ^{k+1} and Δ^{k+1}.
  Increase the iteration counter $k \leftarrow k + 1$ and go to 1.

The algorithm stops either when the poll size parameter falls under a given threshold or when the prescribed budget of function evaluations is spent. Using the Clarke nonsmooth calculus [21], one can prove that under some mild assumptions on the smoothness of the problem, the MADS algorithm globally converges to a solution satisfying local optimality conditions provided that all candidate points lie on the mesh M^k. The interested reader may refer to [8] and [11, Chapter 8].

In the MADS context, surrogates can be used to find new candidate points during the search step. For instance, the few best solutions of a subproblem that uses only surrogates might be some promising candidate points. Several works have tackled the incorporation of surrogates in MADS like quadratic models [22], treed Gaussian processes [30], LOWESS models [60], hybrid models between static surrogates and dynamic models [7], or ensembles of model [13].

3 Quantifying uncertainty with ensembles of models

Ensembles of models enable to combine several models in the hope of taking advantage of each of them. However, this technique creates an aggregate model that can produce a prediction at any point but not an uncertainty on that prediction, thus prohibiting any Bayesian-like approach. Yet, because several models are useful to describe a single function, it means that their predictions are not identical. Consequently, there should be areas in the search space where the predictions show discrepancies, thus resulting in some form of uncertainty that is not apparent in the aggregate prediction. The proposed approach is precisely to catch the discrepancies between the models in order to produce a measure of uncertainty.

The idea that the disparity of the models' predictions can be used is tackled in [29] where the uncertainty at a given point x is produced with the standard deviation between the predictions defined

by

$$\sigma(x) = \left(\frac{\sum_{p=1}^{s} \left(\tilde{f}^p(x) - \bar{f}(x) \right)^2}{s - 1} \right)^{\frac{1}{2}}$$

where $\bar{f}(x) = \sum_{p=1}^{s} \tilde{f}^p(x)/s$. This metric quantifies the gaps between the values of the models, however, as it was said earlier, in a BBO context the actual values matter less than the variations of the models. For instance, the two following models possess significantly different values: $\tilde{f}^1$ and $\tilde{f}^2 = \tilde{f}^1/10 + 20$. Yet, their variations are the same, i.e., when $\tilde{f}^1$ increases, $\tilde{f}^2$ increases too and reciprocally, and therefore they have the same optima. In this case the uncertainty shall be minimum since using either model will yield the same candidate points. Now the two following models differ: $\tilde{f}^1$ and $\tilde{f}^3 = -\tilde{f}^1$, but in addition their variations will be opposite so that their optima will certainly be different. In this case, the uncertainty shall be maximum even though the actual standard deviation between $\tilde{f}^1$ and $\tilde{f}^3$ might be less than between $\tilde{f}^1$ and $\tilde{f}^2$. In light of this, a measure of uncertainty suited to BBO should rather take into account the variations of the models in the form of some local correlation. In [45] and [63], a correlation coefficient is built between a high-fidelity surrogate $\tilde{f}^{\text{high}}$ and a low-fidelity model $\tilde{f}^{\text{low}}$

$$r = \frac{\sum_{j=1}^{M} \left(\tilde{f}^{\text{high}}\left(x^{(j)}\right) - \bar{f}^{\text{high}} \right) \left(\tilde{f}^{\text{low}}\left(x^{(j)}\right) - \bar{f}^{\text{low}} \right)}{\sqrt{\sum_{j=1}^{M} \left(\tilde{f}^{\text{high}}\left(x^{(j)}\right) - \bar{f}^{\text{high}} \right)^2} \sqrt{\sum_{j=1}^{M} \left(\tilde{f}^{\text{low}}\left(x^{(j)}\right) - \bar{f}^{\text{low}} \right)^2}}$$

where $\{x^{(j)}\}_{j \in \{1,2,\ldots,M\}}$ is a set of $M \geq n$ points sampled locally around an area of interest; and $\bar{f}^{\text{high}}$ and $\bar{f}^{\text{low}}$ are the average values of $\tilde{f}^{\text{high}}$ and $\tilde{f}^{\text{low}}$ on this set of points, respectively. For the reasons aforementioned, this quantity is more relevant than the standard deviation in BBO.

3.1 A new expression for the uncertainty

In the proposed approach, this idea of correlation between models is exploited to produce an expression of the uncertainty at a given point x. Two alternatives are built: a smooth and a nonsmooth uncertainties. In addition each alternative is declined into two versions: an uncertainty dedicated to the objective and another one dedicated to the constraints.

Smooth uncertainty for the objective

The simplex gradient [35] of a function f at point x, denoted by $\nabla_S f(x)$, is the gradient of the linear model of f at x. Computing the simplex gradient around a given point x requires the evaluation of f on a simplex, i.e., a set of $n + 1$ affinely independent points, around x. The correlation between two models can be reinterpreted geometrically. For two models $\tilde{f}^p$ and $\tilde{f}^q$, the cosine between their simplex gradients at a given point x is defined by

$$\cos \left\langle \nabla_S \tilde{f}^p(x), \ \nabla_S \tilde{f}^q(x) \right\rangle = \frac{\nabla_S \tilde{f}^p(x)^\top \nabla_S \tilde{f}^q(x)}{||\nabla_S \tilde{f}^p(x)||_2 \times ||\nabla_S \tilde{f}^q(x)||_2} \ .$$

The larger the cosine, the more correlated the models around x. With this notion in mind, the uncertainty between two models $\hat{\sigma}_{p,q}$ can be produced as an inversely proportional function of the cosine

$$\hat{\sigma}_{p,q}(x) := \frac{1}{2} \left(1 - \cos \left\langle \nabla_S \tilde{f}^p(x), \ \nabla_S \tilde{f}^q(x) \right\rangle \right) \ . \tag{2}$$

When the models are highly correlated, the cosine is close to 1 so that the uncertainty is close to its minimum 0. When the models are poorly correlated, the cosine is closer to 0 and the uncertainty increases to 0.5. And when the models are anti-correlated, the cosine is close -1 so that the uncertainty reaches its maximum 1. The choice of a simplex is left at the discretion of the user. A small simplex

around x will yield a simplex gradient that is a good approximation of the true gradient for smooth functions, but a wider simplex will have a smoothing effect that can be appreciable with nonsmooth or noisy functions. In the current context, the simplex gradient acts as a simple surrogate for the true gradients of the models $\tilde{f}^p$, $p \in \{1, 2, \ldots, s\}$ that are not always easy to obtain. However, using the true gradients if available might be equally efficient. See Appendix A for the practical construction of the simplex used in this work.

The generalization of this expression to more than two models will be described after the other versions of uncertainties are introduced.

Nonsmooth uncertainty for the objective

An alternative for the uncertainty that does not require the computation of simplex gradients is proposed. It requires a positive spanning set of directions $\mathcal{D}$, i.e., a set of at least $n + 1$ vectors of $\mathbb{R}^n$ such that any point of $\mathbb{R}^n$ can be written as a positive linear combination thereof [26]. The nonsmooth alternative is defined by

$$\hat{\sigma}_{p,q}(x) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \mathsf{xor}\Big(\tilde{f}^p(x + d) < \tilde{f}^p(x) \ , \ \tilde{f}^q(x + d) < \tilde{f}^q(x) \Big) \tag{3}$$

where $\mathsf{xor}(\cdot \, , \cdot)$ is the exclusive or logical operator. For each direction $d \in \mathcal{D}$, the uncertainty increases if the models predict contradictory trends, i.e., if model $\tilde{f}^p$ increases from x to $x + d$ while $\tilde{f}^q$ decreases, or conversely. The term $1/|\mathcal{D}|$ scales the sum between 0 and 1 so that the uncertainty will not be influenced by the number of directions in $\mathcal{D}$. Here again, the choice of a positive spanning set is left at the discretion of the user. Not only can the size of the directions vary as with the simplex, but also the number of directions can increase in order to explore the surroundings of x better. See Appendix A for the practical construction of the positive spanning set used in this work.

Smooth uncertainty for the constraints

The expressions for the uncertainty proposed in Equations (2) and (3) are suited for the objective f since they take into account variations of the models. However, when handling constraints, the key information is the sign of the function rather than whether it increases or not. If two models $\tilde{c}_j^p$ and $\tilde{c}_j^q$ of the same constraint c_j are available, the uncertainty shall increase when one model or the other tends towards 0, and increase even more when their signs are opposite, meaning that their predictions on the feasibility are contradictory. Hence the following expression

$$\hat{\sigma}_{p,q}(x) = \mathrm{sigm}\left(-\tilde{c}_j^p(x) \times \tilde{c}_j^q(x) \right) \tag{4}$$

where $\mathrm{sigm}(\cdot)$ is the sigmoid function. It acts as an activation function that increases as the product of $\tilde{c}_j^p$ and $\tilde{c}_j^q$ decreases. This uncertainty also ranges from 0 to 1.

Nonsmooth uncertainty for the constraints

Here again, a nonsmooth alternative is proposed to the smooth uncertainty for the constraints. It uses the logical operator xor to indicate whether the two models $\tilde{c}_j^p$ and $\tilde{c}_j^q$ predict the same feasibility result at a given point x or not

$$\hat{\sigma}_{p,q}(x) = \mathsf{xor}\left(\tilde{c}_j^p(x) \leq 0 \ , \ \tilde{c}_j^q(x) \leq 0 \right) \tag{5}$$

Generalization to an arbitrary number of models

Expressions (2), (3), (4) and (5) consider two models of the objective or a constraint. Ensembles of models usually comprise more than two models though, hence the need for a general expression that can consider an arbitrary number of models. In addition, this general expression must take into account the weights w^p, $p \in \{1, 2, \ldots, s\}$, which reflect the quality of the models. Just as in the prediction

defined in Equation (1), the good models should have a strong influence in the determination of the uncertainty whereas the poor models should not. The following quantity meets those requirements

$$\left(\sum_{p=1}^{s-1}\sum_{q=p+1}^{s} w^p w^q \times \hat{\sigma}_{p,q}(x)\right) \bigg/ \sum_{p=1}^{s-1}\sum_{q=p+1}^{s} w^p w^q \qquad (6)$$

The ratio in (6) considers all the possible pairs of models once. For each pair $(p,q) \in \{1,2,\ldots,s\}^2$ such that $p \neq q$, the uncertainty $\hat{\sigma}_{p,q}(x)$ stemming from the models $\tilde{f}^p$ and $\tilde{f}^q$ at point x is weighted by the product of the corresponding weights w^p and w^q. Consequently, the better the models, the more $\hat{\sigma}_{p,q}(x)$ will weight up in the total uncertainty at point x. Then the sum on all pairs of models is normalized by $1/\sum_{p=1}^{s-1}\sum_{q=p+1}^{s} w^p w^q$ so that the result does not depend on the number of models s. Since the four versions of $\hat{\sigma}_{p,q}$ range from 0 to 1, this ratio applies to any case: objective and constraint versions, smooth and nonsmooth alternatives.

At this point the uncertainty takes into account an arbitrary number of models and also the weights as required. But the ratio in (6) is between 0 and 1 by construction, and therefore it is most likely not at the right scale for the problem at hand. A final step is to multiply by a factor $\alpha > 0$ that scales the ratio in a relevant way. For this purpose, $\alpha = 10 \times \text{Var}(g(V))$ was chosen, where g is either the objective or a constraint; $g(V) = \{g(x^{(1)}), g(x^{(2)}), \ldots, g(x^{(N_s)})\}$ is the set of already sampled values of the function g; and 10 is a factor that empirically gave better results. This choice is motivated by the fact that the problem's scale can only be known through the true function's values. The final expression of the uncertainty is

$$\hat{\sigma}(x) = \alpha \frac{w^\top \Sigma(x) w}{w^\top T w} \qquad (7)$$

where the ratio (6) has been rewritten in a more compact form with $\Sigma(x) \in \mathbb{R}^{s \times s}$ being the upper triangular matrix such that $[\Sigma(x)]_{p,q} = \hat{\sigma}_{p,q}(x)$ if $p < q$ and 0 otherwise; $T \in \mathbb{R}^{s \times s}$ being the upper triangular matrix such that $[T]_{p,q} = 1$ if $p < q$ and 0 otherwise; and w being the vector of weights $[w^1, w^2, \ldots, w^s]^\top$.

The different uncertainties are illustrated in Figure 2. Seven points have been sampled in $[-10, 10]^2$ from an unknown function g that takes two input variables x_1 and x_2. On each subfigure, the bottom surface is the aggregate prediction $\hat{g}$ resulting from eleven different polynomial and RBF models, and the top surface, shifted up for readability, is the uncertainty on that prediction which becomes darker as it increases. The weights of the models w^p, $p \in \{1, 2, \ldots, s\}$, are attributed as described in Section 3.2. In Figures 2a and 2b the function is interpreted as the objective whereas in Figures 2c and 2d the same function is interpreted as a constraint, resulting in significantly different uncertainties. In addition, in Figures 2c and 2d, the points close to the assumed border of the constraint, i.e., where the prediction is close to zero, were darkened in order to better understand the uncertainty.

Comparatively, Figure 3 shows a GP model's prediction and uncertainty fit on the same sample points. It can be noticed that the uncertainty is lower close to the sample points and increases with the distance to them, which is expected with a GP model. The same observation cannot be made in Figure 2, especially in Figures 2c and 2d where the uncertainty is higher close to the border of the constraint.

3.2   Error metric and weight attribution

In this work, the strategy to assign weights is to compute an error metric $\mathcal{E}^p$ for each model $\tilde{f}^p$ and then attribute a weight w^p that is a function of $\mathcal{E}^p$. The error metric chosen is the order error cross-validation metric (OECV) [13] mentioned earlier and denoted by $\mathcal{E}_{OECV}$. Broadly speaking, it measures a model capacity to rank points in the same order as the actual objective would do, or, for that matter, its capacity to predict the same feasibility result as an actual constraint would do. When the metric $\mathcal{E}_{OECV}^p$ is computed for every model $\tilde{f}^p$, $p \in \{1, 2, \ldots, s\}$, the weights can be attributed. Assigning a weight of 1 to the best model and 0 to the others is the choice made in [13]. However, in the present work there must be at least two strictly positive weights otherwise the ratio (6) is a division by zero

(a) Objective version - smooth alternative.

(b) Objective version - nonsmooth alternative.

(c) Constraint version - smooth alternative.

(d) Constraint version - nonsmooth alternative.
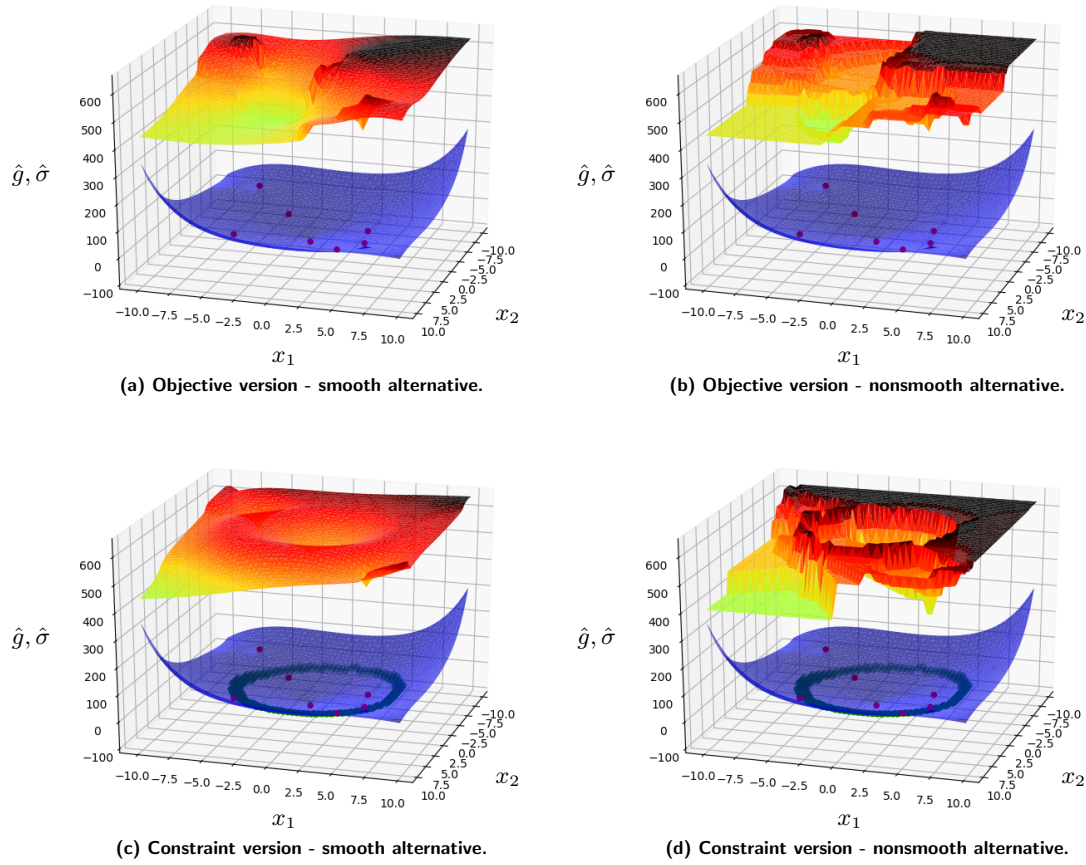
Figure 2: The four uncertainties on the same sample set. Figures **2a** and **2b** correspond to the the smooth and nonsmooth alternatives of the objective version, respectively (Equations (2) and (3)). Figures **2c** and **2d** correspond to the the smooth and nonsmooth alternatives of the constraint version, respectively (Equations (4) and (5)).
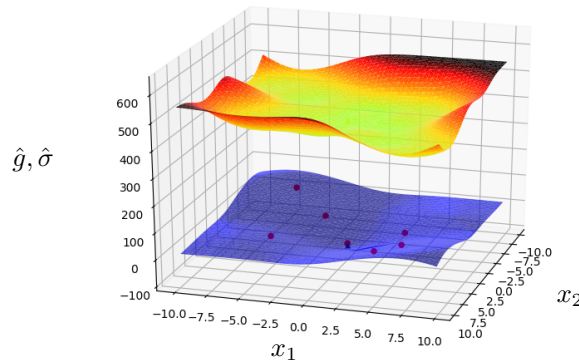
Figure 3: Prediction and uncertainty of Gaussian processes.

and has no meaning. The approach chosen instead is to select the N_{best} models that have the smallest error metrics and to assign to them a weight proportional to the metric. Formally, if $I \subset \{1, 2, \ldots, s\}$ is the subset of the selected models indices, then $w^p \propto \mathcal{E}_I^{\text{tot}} - \mathcal{E}^p$ if $p \in I$ and $w^p = 0$ otherwise, where $\mathcal{E}_I^{\text{tot}}$ is the total error of the selected models. The weights are then normalized so that $\sum_{p \in I} w^p = 1$.

If more than N_{best} models have an error metric that is equal to the best metric, all of them will be selected and be assigned an equal weight. Preliminary tests showed that $N_{\text{best}} = 3$ and $N_{\text{best}} = 4$ were appropriate values for the smooth and nonsmooth alternatives, respectively.

3.3 Incorporation into the MADS algorithm

The MADS algorithm offers an important flexibility through its search step. Many works have already included model-based subproblems (SP) into the search step as described earlier. The proposed implementation falls within this category. At each iteration, a surrogate problem is solved during the search step and the best solution is used as the next candidate point for the true problem. Having said that, there are many ways to design a surrogate SP.

In [61], eight different formulations for SP are proposed, denoted by SP1 through SP8. They are specifically designed to take advantage of the double information that stochastic models provide, i.e., the prediction and the uncertainty. These formulations tackle general constraints and involve the following statistical measures:

$$
\left\{
\begin{array}{rrcl}
\text{Expected improvement [34]:} & \text{EI}(x) & = & \mathbb{E}[\max(f_{min} - f(x), 0)] \\
\text{Probability of feasibility [61]:} & \text{P}(x) & = & \mathbb{P}[c_j(x) \leq 0, j = 1, 2, \ldots, m] \\
\text{Probability of improvement [33]:} & \text{PI}(x) & = & \mathbb{P}[f_{min} > f(x)]
\end{array}
\right.
$$

Then some other measures are derived from these quantities: the *expected feasible improvement* $\text{EFI}(x) = \text{EI}(x)\text{P}(x)$, the *probability of feasible improvement* $\text{PFI}(x) = \text{PI}(x)\text{P}(x)$ and the *uncertainty on the feasibility* $\mu(x) = 4\text{P}(x)(1 - \text{P}(x))$. In the formulations, all these measures are arranged in different ways in order to highlight various properties of the sample points. The eight formulations of SP are given in Appendix B.

With stochastic models, the probability distribution at any point x is known and used to practically compute $\text{EI}(x)$, $\text{P}(x)$ and $\text{PI}(x)$. However, an aggregate model extended with the proposed uncertainty, although inspired by the stochastic modelling philosophy, has no probabilistic foundation. More specifically, there is no cumulative distribution function available at a given point $\mathbb{P}[g(x) < g_0]$, for all $g_0 \in \mathbb{R}$. Consequently, the statistical quantities defined above are not applicable as such. To address this issue, P, PI and EI are replaced by substitutes $\widetilde{\text{P}}$, $\widetilde{\text{PI}}$ and $\widetilde{\text{EI}}$ that are inspired from the case when the stochastic model yields at any point x a value that follows a normal distribution $\mathcal{N}\big(\hat{y}(x), \hat{\sigma}^2(x)\big)$, which is the case of GPs.

When $\hat{c}_j(x) \sim \mathcal{N}\big(\hat{y}_j, \hat{\sigma}_j^2\big)$, the expression of P becomes

$$
\text{P}(x) = \prod_{j=1}^{m} \Phi\left(-\frac{\hat{y}_j}{\hat{\sigma}_j}\right)
$$

where Φ is the normal cumulative distribution function. Here the product implies that the constraints are assumed to be independent from each other, which might be incorrect but it is the best available approximation in a BBO context. The proposed adaptation is

$$
\widetilde{\text{P}}(x) = \prod_{j=1}^{m} \text{sigm}_\lambda\left(-\frac{\hat{y}_j}{\hat{\sigma}_j}\right)
$$

where sigm_λ is the sigmoid function of parameter λ, i.e., $\text{sigm}_\lambda(x) = \text{sigm}(\lambda x)$. Both Φ and sigm_λ tend to 1 when the ratio $-\hat{y}_j/\hat{\sigma}_j$ tends to $+\infty$, i.e., either when $\hat{y}_j$ is highly negative or when the uncertainty is low for whatever negative value of $\hat{y}_j$, which in both cases means that the constraint c_j is most likely satisfied. They also tend to 0 when the ratio $-\hat{y}_j/\hat{\sigma}_j$ tends to $-\infty$, i.e., either when $\hat{y}_j$ takes high values or when the uncertainty is low for whatever positive value of $\hat{y}_j$, which in both cases means that the constraint c_j is most likely not satisfied. $\lambda = 2$ produces the closest approximation of Φ

but choosing other values enable to control the shape of the function. Preliminary tests showed that $\lambda = 3$ and $\lambda = 1$ were interesting values for the smooth and nonsmooth alternatives, respectively.

As for PI, when $\hat{f}(x) \sim \mathcal{N}(\hat{y}, \hat{\sigma}^2)$, the expression becomes

$$\text{PI}(x) = \Phi\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

where f_{min} is the best know value of the objective. The proposed alternative is

$$\widetilde{\text{PI}}(x) = \text{sigm}_\lambda\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

Here again, Φ and sigm_λ have the same behaviour but the parameter λ enables to control the shape of PI. The values chosen for the smooth and nonsmooth alternatives are $\lambda = 0.1$ and $\lambda = 0.5$, respectively.

Finally, when $\hat{f}(x) \sim \mathcal{N}(\hat{y}, \hat{\sigma}^2)$, the expression of EI becomes

$$\text{EI}(x) = (f_{min} - \hat{y})\Phi\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right) + \hat{\sigma}\phi\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

where ϕ is the normal density function. This expression is intimately related to the Gaussian aspect of the model an therefore is not *a priori* suited for non-Gaussian models, let alone models that are not truly stochastic. However, it possesses interesting properties that are independent from the Gaussian nature of the model and that can be seen as essential to the very notion of expected improvement. Firstly, for a fixed $\hat{\sigma}$, EI decreases in $\hat{y}$, tends to 0 when $\hat{y}$ tends to $+\infty$ and is almost proportional to $\hat{y}$ when $\hat{y}$ tends to $-\infty$, which is judicious in a minimization context. In addition, when $\hat{y}$ gets closer to f_{min}, EI becomes almost proportional to $\hat{\sigma}$, meaning that when the prediction does not improve the objective (i.e., $\hat{y} \simeq f_{min}$) the expected improvement mostly relies on the uncertainty. Then, for a fixed $\hat{f}$, EI increases in $\hat{\sigma}$ and is almost proportional to $\hat{\sigma}$ when $\hat{\sigma}$ tends to $+\infty$, which is sensible since for a given prediction the higher the uncertainty, the larger the potential improvement. Finally, when $\hat{\sigma}$ tends to 0, the behavior of EI depends on the values of f_{min} and $\hat{y}$: if $f_{min} \geq \hat{y}$, then EI tends to $f_{min} - \hat{y}$, and if $f_{min} < \hat{y}$, then EI tends to 0, meaning that when the uncertainty is low, EI mostly relies on the comparison between f_{min} and the prediction $\hat{y}$. Taking into account these considerations, the proposed adaptation for EI is very close to the actual EI and is defined by

$$\widetilde{\text{EI}}(x) = (f_{min} - \hat{y})\,\text{sigm}_\lambda\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right) + \hat{\sigma}\gamma\left(\frac{f_{min} - \hat{y}}{\hat{\sigma}}\right)$$

where $\gamma(t) = e^{-t^2/2}$. Here $\lambda = 1$ was chosen. The functions ϕ and γ only differ by a factor $1/\sqrt{2\pi}$ and the reason for choosing γ instead of ϕ is that this factor is no more justified without an actual stochastic model that produces normal distributions. Moreover, preliminary tests showed that the proposed uncertainty seemed on average lower than the uncertainty provided by a kriging model. The terms P, PI and EI were then replaced by $\widetilde{\text{P}}$, $\widetilde{\text{PI}}$ and $\widetilde{\text{EI}}$ in the formulations of SP.

Algorithm 3 summarizes the incorporation of extended aggregate models in MADS. First, one formulation must be chosen among {SP1, SP2, ..., SP8}. At iteration k, the best solution found for SP, denoted by x_{SP}^k, is projected onto the mesh M^k and is used as the candidate point of the search step. The freshly evaluated points are then added to the cache V^k so that the models and the weights will be adjusted accordingly before iteration $k + 1$ begins. The resulting algorithm benefits from the convergence results of MADS since all the candidate points lie on the mesh M^k.

4   Computational results

The proposed approach has been tested on seven analytical problems; two multi-disciplinary optimization (MDO) applications: the aircraft range problem and the simplified wing problem; and two simulation

---

Algorithm 3: The MADS algorithm with aggregate models.

---

0. Initialization

   $\text{SP} \in \{\text{SP}_1, \text{SP}_2, \ldots, \text{SP}_8\}$: surrogate subproblem formulation
   $\tilde{g}^1, \tilde{g}^2, \ldots, \tilde{g}^s$: choice of models for the objective and the constraints
   $V^0 \subset \mathbb{R}^n$: set of starting points
   $\Delta^0 \geq \delta^0 > 0$: initial mesh and poll size parameters
   $k \leftarrow 0$: iteration counter

1. Models and weights update

   Build or update $\tilde{f}^1, \tilde{f}^2, \ldots, \tilde{f}^s$ using the values of f in V^k
   Update $w^1, w^2, \ldots, w^s$ using the OECV metric for the objective
   Build or update $\tilde{c}_j^1, \tilde{c}_j^2, \ldots, \tilde{c}_j^s$ using the values of c_j in V^k, for $j \in \{1, 2, \ldots, m\}$
   Update $w_j^1, w_j^2, \ldots, w_j^s$ using the OECV metric for the constraints, for $j \in \{1, 2, \ldots, m\}$

2. Search

   Solve SP to find the best solution x_{SP}^k
   Project x_{SP}^k onto the mesh M^k
   Evaluate the resulting point with the true problem

3. Standard poll

4. Standard parameters update

---

problems: solar1 and styrene. Version 4 of the NOMAD software [1, 14] was used on a PC Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz on Linux. The aggregate models used the default selection of eighteen models comprised of polynomial response surfaces of various degrees, kernel smoothing, modified radial basis functions as in [13], and closest neighbours. The competing quadratic and kriging models were also readily available in NOMAD. Due to the long running times required by the kriging models, the latter were only tested on the analytical problems and the aircraft range problem. Every instance of MADS in this work uses the last direction of success at the poll step [8]. Two other BBO solvers have been included in this study: DFN [27] and SHEBO [46]. The former innately handles general constraints whereas the latter is designed for problems with hidden constraints. Consequently, the problems were adapted in SHEBO so that a violated general constraint will be interpreted as a hidden constraint.

The interpretation of the results mostly relies on data profiles [44] which enable to compare multiple solvers on a given set of problems. Broadly speaking, the data profile of a solver indicates the proportion of problems solved to a given tolerance within a prescribed number of evaluations. Since SHEBO does not take a single starting point as an input, it is not fit for comparison to the other algorithms through data profiles. Section 4.6 provides tabular comparisons with SHEBO.

Unless otherwise specified, due to the randomness contained in MADS, every version thereof was run four times on each problem with a different seed for the random generator each time. Similarly, DFN enables to choose between the Halton and Sobol sequences so the two were tested on each problem and taken into account in the data profiles.

4.1 Analytical problems

The seven analytical problems are listed in Table 1 with the number of variables n and constraints m, whether the variables are bounded or not, and the number of starting points used. By taking into account the additional starting points for problems HS83, HS114 and MAD6, the total number of problems is fifteen. The evaluation budget is $1200(n + 1)$.

As in [61], the eight SP formulations were compared, and the following values were tested for the parameter λ when applicable: $\{0, 0.01, 0.1, 1\}$, thus resulting in twenty-three distinct formulations. When this parameter is involved in a formulation, it is denoted as a subscript, e.g., $\text{SP2}_{0.1}$. The purpose here is not to exhaustively compare the formulations with each other but rather to identify the bests formulations and compare their performances to the existing versions of NOMAD and to DFN.

Table 1: Description of the seven analytical problems.

#	Name	Source	n	m	Bounds	# starting points
1	G2	[10]	10	2	yes	1
2	HS19	[32]	2	2	yes	1
3	HS83	[32]	5	6	yes	4
4	HS114	[40]	9	4	yes	3
5	MAD6	[40]	5	7	no	4
6	PENTAGON	[40]	6	15	no	1
7	SNAKE	[9]	2	2	no	1

The formulations with extended aggregate models were compared to NOMAD without any search step, referred to as "no search", NOMAD with a search step involving the minimization of quadratic models, referred to as "quad search", and DFN. It turns out that all formulations perform better than no search, confirming that the approach is valid and does not "waste" evaluations. However, quad search is most of the times as good as, and sometimes better than, the proposed approach, confirming the effectiveness of quadratic models on analytical functions. DFN presents heterogeneous performances. It found good solutions for three problems but performed poorly on the others, hence the low overall performance.

The best formulation found for the smooth alternative in terms of the proportion of problems solved within the budget is $SP3_0$ defined by

$$\min_{x \in \mathcal{X}} \quad -\mathrm{EI}(x)$$
$$\text{s.t.} \quad \hat{c}_j(x) \leq 0, \quad j = 1, 2, \ldots, m \tag{SP3-EIσ}$$

As for the nonsmooth alternative, the best formulation is $SP5_{0.01}$ which consists in maximizing $\mathrm{EFI}(x) + 0.01\hat{\sigma}_f(x)$. Figure 4 shows the data profiles of the five following algorithms: no search, quad search, $SP3_0$ with smooth uncertainty, $SP5_{0.01}$ with nonsmooth uncertainty, and DFN at variable tolerance: $\tau = 10^{-1}$, $\tau = 10^{-3}$, $\tau = 10^{-5}$ and $\tau = 10^{-7}$.

The profiles show that the performances of SP3 and SP5 are close to that of quad search for high tolerance. However, quad search becomes significantly better for low tolerance ($\tau = 10^{-7}$). These first results show that the extended aggregate models combined with the formulations manage to find good solutions as efficiently as quad search. However, quadratic models do so slightly faster in terms of the number of evaluations, and more importantly, they are especially accurate on analytical problems, thus resulting in superior performances at low tolerance.

Since extended aggregate models are meant to mimic and therefore supersede actual stochastic models, the comparison to the available kriging models in NOMAD is most appropriate. In [61], the authors recommend SP1 and SP2 with large values of λ when using stochastic models. For that reason, these two formulations have been tested with $\lambda = 0.1$ and $\lambda = 1$. On this set of problems, $SP2_{0.1}$ turns out to be the best formulation. The latter was therefore tested against $SP3_0$ with smooth uncertainty and $SP5_{0.01}$ with nonsmooth uncertainty, that is the best formulations seen above. The resulting data profiles in Figure 5 show that on the present set of problems extended aggregate models are as good as, or better than, the kriging alternative depending on the tolerance. In addition, due to the inversion of a covariance matrix that grows with the size of the sample set, the kriging models typically take minutes to tens of minutes to solve one problem, which is prohibitive when optimizing cheap functions with large budgets of evaluations. In comparison, the proposed approach and quad search typically take minutes and no search and DFN take seconds. As a result, replacing classic kriging models by extended aggregate models does not harm the performances in terms of the number of evaluations on the set of analytical problems, while improving the real optimization time. Overall, based on the present results, quad search must be favoured on cheap analytical problems.
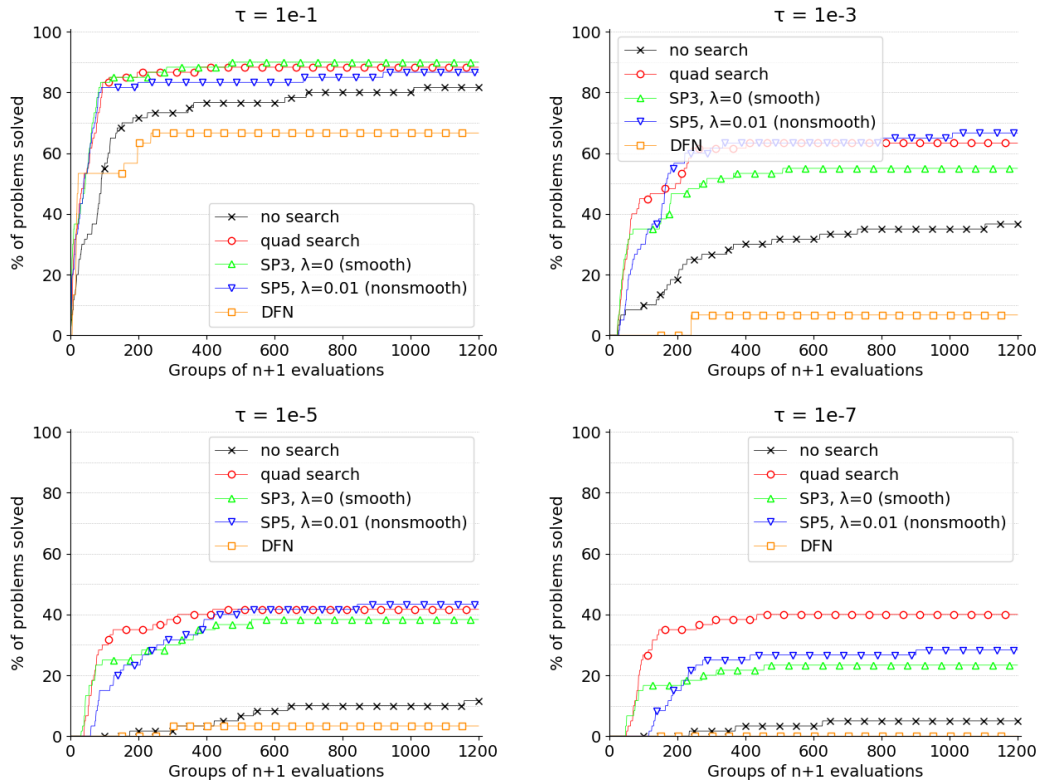
Figure 4: Data profiles. no search vs. quad search vs. SP3$_0$ with smooth uncertainty vs. SP5$_{0.01}$ with nonsmooth uncertainty vs. DFN on analytical problems.

## 4.2 The aircraft range MDO problem

The aircraft range problem [37] is a multi-disciplinary optimization problem (MDO), meaning that it is combined of several interconnected disciplines so that the input of one discipline is the output of the others, and several cycles between them are necessary to stabilize the result. The aircraft range problem aims at maximizing the range of a supersonic business jet by considering aerodynamics, structure, and propulsion, under constraints of engine, performance and structure. The problem has $n = 10$ variables and $m = 10$ constraints. It is nonsmooth and has several local optima.

In order to test the algorithms truthfully, ten starting points were sampled with Latin hypercube sampling [42]. Here again, the 23 formulations were run with the two uncertainty alternatives in order to identify the best combinations and analyze their performances. The evaluation budget is $1000(n+1)$. The best formulation with the smooth alternative is SP3$_{0.1}$ defined by

$$\min_{x \in \mathcal{X}} \; -\text{EI}(x) - 0.1\hat{\sigma}_f(x)$$
$$\text{s.t.} \quad \hat{c}_j(x) - 0.1\hat{\sigma}_j(x) \le 0, \quad j = 1, 2, \ldots, m \tag{SP3-EIσ}$$

and the best formulation identified with the nonsmooth alternative is SP8 which consists in maximizing PFI(x).

This time, each of the aforementioned formulations is compared individually not only to quad search, but also to the same formulation with kriging models instead. This choice is motivated by the fact that the aircraft range problem is a real-world simulation-based problem that takes significantly more time to compute than the previous analytical problems, and therefore computing expensive kriging models might be worth the trade-off. In this section, no search is not shown in the interest of readability, and DFN neither because it performed poorly and its data profiles were flat. Figures 6a and 6b show the data
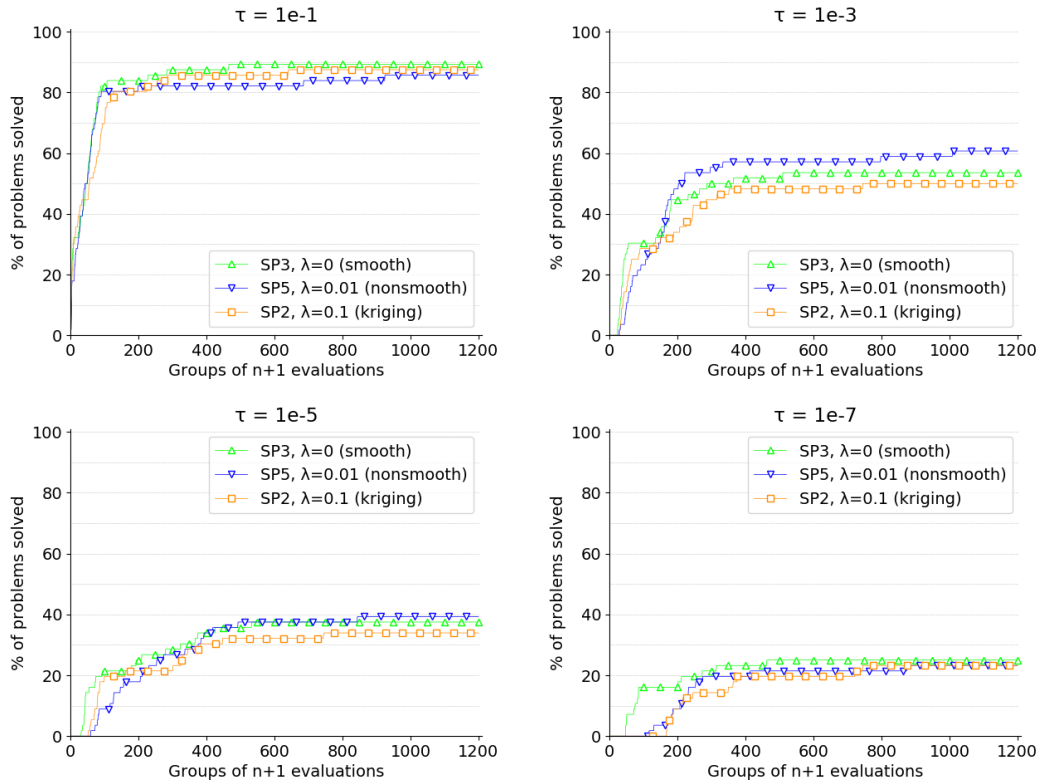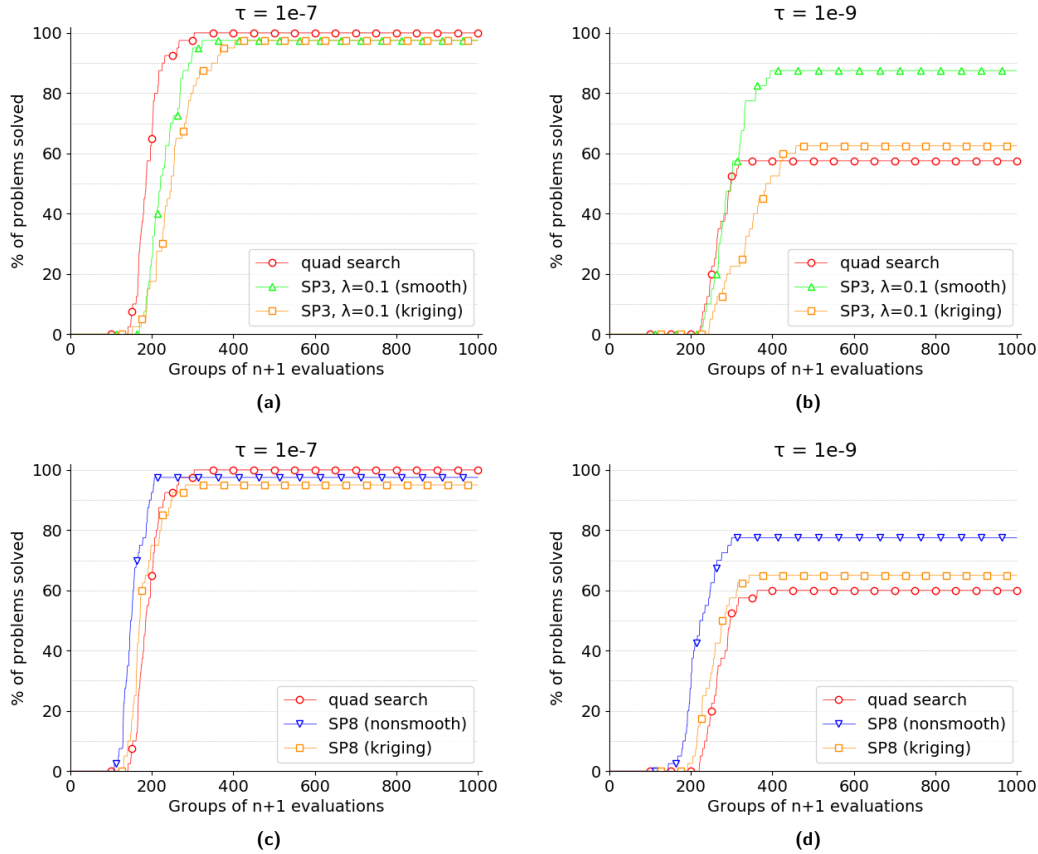
Figure 5: Data profiles. $SP3_0$ with smooth uncertainty vs. $SP5_{0.01}$ with nonsmooth uncertainty vs. $SP2_{0.1}$ with kriging models on analytical problems.

profiles of $SP3_{0.1}$ with smooth uncertainty with tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, respectively; and Figures 6c and 6d show the data profiles of SP8 with nonsmooth uncertainty with tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, respectively. All the algorithms presented equivalent performances for high tolerances ($\tau = 10^{-1}$ and $\tau = 10^{-3}$) and consequently the data profiles did not reveal significant difference between the solvers with tolerances under $\tau = 10^{-5}$, meaning that all the formulations manage to reach a good solution.

For every above-mentioned formulation, the performance is relatively comparable to that of quad search for tolerance $\tau = 10^{-7}$. However, for $\tau = 10^{-9}$, the proposed extended aggregate models coupled with the right formulations turn out to be significantly better than both quad search and their kriging counterpart, i.e., the same formulations with kriging models instead. It can be noticed that the smooth alternative eventually solves more problem, but the nonsmooth one solves problems faster. In Figure 6b, $SP3_{0.1}$ with smooth uncertainty solves 87.5% of the problems at tolerance $\tau = 10^{-9}$ with the allocated budget, while in Figure 6d SP8 with nonsmooth uncertainty only solves 77.5% of the problems at the same tolerance. However, the latter takes only $300(n+1)$ evaluations to do so, while the former has only solved 50% of the problems after the same number of evaluations. This trend has been observed on this problem with all the other formulations not presented in this paper. The higher achievements of the smooth uncertainty over the long run may be attributed to its intrinsically rich range of values, while the relative rapidity of the nonsmooth uncertainty may be the result of a more aggressive behaviour that helps find a good solution faster.

It could be argued that comparing kriging models within the best formulations found for extended aggregate models, i.e., $SP3_{0.1}$ and SP8, is not fair since the former might perform poorly on these formulations but yield better results on others. In [61], the authors use stochastic models and recommend SP5, SP6 and SP7 with small values of λ for expensive simulation-based problems as is the case with

the aircraft range problem. Accordingly, those three formulations were tested with kriging models and $\lambda = 0.01$. On this particular problem, SP5 turns out to be the best formulation. Consequently, extended aggregate models were compared to kriging models on SP5. Figures 7a and 7b show the data profiles of $SP5_{0.01}$ with smooth and nonsmooth alternatives, and kriging models. The latter perform indeed better than the extended aggregate models on $SP5_{0.01}$ at tolerances $\tau = 10^{-7}$ and $\tau = 10^{-9}$, meaning that the best formulations with real stochastic models are not necessarily the same than the best ones with the proposed extended aggregate models.

Figure 6: Data profiles. Figures 6a and 6b show quad search vs. $SP3_{0.1}$ with smooth uncertainty vs. $SP3_{0.1}$ with kriging models. Figures 6c and 6d show quad search vs. SP8 with nonsmooth uncertainty vs. SP8 with kriging models on the aircraft range problem.

As a result, the fair comparison is not between kriging models and extended aggregate models *within the same formulation*, but rather between each type of models coupled with its best formulation, that is $SP3_{0.1}$ with the smooth alternative, SP8 with the nonsmooth alternative, and SP5 with kriging models. Figures 7c and 7d show the data profiles of the above combinations and in addition quad search as a reference. At tolerance $\tau = 10^{-7}$, the performances are alike but at tolerance $\tau = 10^{-9}$, the extended aggregate models coupled with the suitable formulations solve more problems, faster than kriging models with their own appropriate formulation, and faster than quad search. To tolerance $\tau = 10^{-9}$, the smooth and nonsmooth alternatives solve 87.5% and 75% of the problems, respectively, while kriging models and quad search solve 72.5% and 57.5% of the problems, respectively.

Since the different approaches require some non negligible amount of internal computation, it is appropriate to compare them not only in terms of the number of evaluations, but also in terms of the total real optimization time. It was highlighted in Section 4.1 that kriging models were prohibitively long to train when optimizing cheap analytical problems. The same question is addressed more thoroughly with the aircraft range problem. Figure 8 shows the *time data profiles* of formulation $SP3_{0.1}$ with
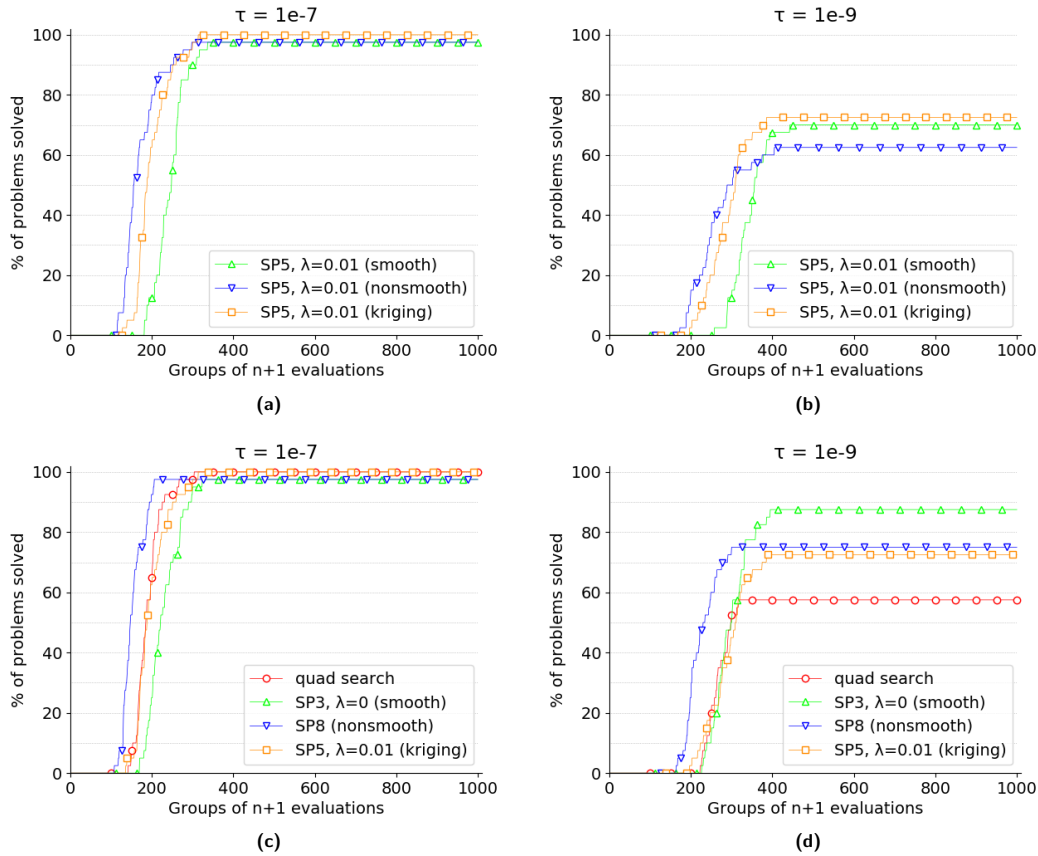
Figure 7: Data profiles. Figures 7a and 7b show $SP_{0.01}$ with smooth uncertainty vs. $SP5_{0.01}$ with nonsmooth uncertainty vs. $SP5_{0.01}$ with kriging models. Figures 7c and 7d show quad search vs. $SP3_{0.1}$ with smooth uncertainty vs. SP8 with nonsmooth uncertainty vs. $SP5_{0.01}$ with kriging models on the aircraft range problem.

the smooth alternative, formulation SP8 with the nonsmooth alternative, and formulation $SP5_{0.01}$ with kriging models. In a time data profile, the proportion of problems solved is not a function of the number of evaluations anymore but a function of the real computation time instead. The profiles suggest that even with a more expensive, real-world problem, kriging models are especially long to train. At tolerance $\tau = 10^{-9}$, $SP3_{0.1}$ with the smooth alternative solves 87.5% of the problems within 95 seconds and SP8 with the nonsmooth alternative solves 75% of the problems within 73 seconds. After the same amount of time, SP5 with kriging models has solved less than 10% of the problems, and requires 728 seconds to solve 65% of the problems. Some instances even require more than 1500 seconds. Based on the present results, the proposed extended aggregate models constitute a cheaper and more efficient alternative to kriging models.

4.3 The simplified wing problem

The simplified wing problem [66] is also an MDO problem. It aims at minimizing the drag of a wing by taking into account aerodynamics and structure. This problem is smooth but has several local optima. It has $n = 7$ bounded variables and $m = 3$ constraints.

Ten starting point have been randomly sampled in the bounded space of variables with Latin hypercube sampling. The evaluation budget is $600(n + 1)$. Unlike the previous problems and for the rest of the study, the kriging models have not been tested because of their heavy computational cost, and not all the formulations have been tested but rather a subset of the most promising ones found on the aircraft range problem: $SP1_1$, $SP2_0$, $SP3_{0.1}$, $SP7_1$ and SP8 for the smooth alternative; and $SP2_{0.1}$,
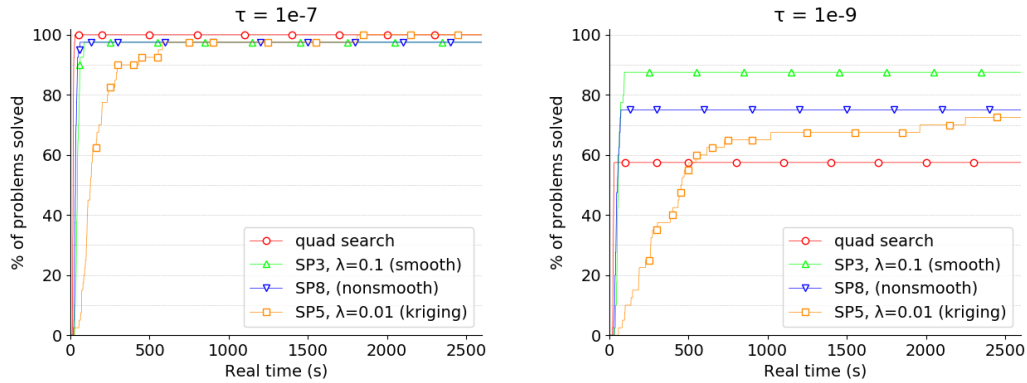
Figure 8: Time data profiles. quad search vs. SP$3_{0.1}$ with smooth uncertainty vs. SP8 with nonsmooth uncertainty vs. SP$5_{0.01}$ with kriging models on the aircraft range problem.

SP$3_{0.01}$, SP4, SP$6_{0.1}$ and SP8 for the nonsmooth alternative. Among the above formulations, the best ones found for this problem with the smooth and nonsmooth alternatives are SP8 and SP4, respectively. SP4 consists in maximizing $\text{EFI}(x)$. Figure 9 shows the data profiles of no search, quad search, SP8 with smooth uncertainty, SP4 with nonsmooth uncertainty, and DFN at tolerances $\tau = 10^{-1}$ and $\tau = 10^{-3}$. Lower tolerances resulted in data profiles that were too flat, especially because of the high sensitivity to the seed of NOMAD on this problem.
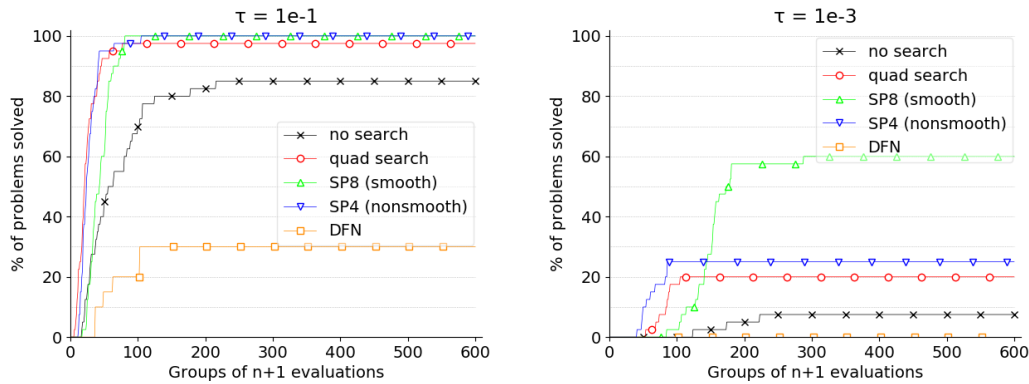
Figure 9: Data profiles. no search vs. quad search vs. SP8 with smooth uncertainty vs. SP4 with nonsmooth uncertainty vs. DFN on the simplified wing problem.

On this MDO problem, the extended aggregated models perform better than no search, quad search and DFN. SP4 with the smooth uncertainty is especially good at tolerance $\tau = 10^{-3}$, solving 60% of problems while quad search only solves 20%. As with the aircraft range problem, the nonsmooth alternative solves problems faster than the smooth one. However, on the simplified wing problem the smooth alternative is by far the most efficient eventually.

4.4   The solar1 problem

The solar1 problem is part of a set of nine concentrated solar power simulation problems [28] that serve as a benchmark for blackbox optimization solvers, available at `github.com/bbopt/solar`. The solar1 problem aims at maximizing the heliostat field energy output throughout one day under constraints of field geometry and cost. The problem is noisy and has several local optima. It has $n = 9$ variables, among which one is discrete, and $m = 5$ constraints. The solar1 problem has the specificity of having two additional adjustable parameters: the seed and the number of replications. Since the problem

contains some stochasticity, the seed for the random generator can be chosen. It is *not* the same seed as the one of NOMAD. Two instances of solar1 with two different seeds are considered as two different problems in this work. The other parameter enables to replicate the evaluations in order to smooth the problem and compensate the noise. It was fixed to ten in the experiments.

In order to generate several problem instances, fifteen different seeds are chosen for the problem - not for NOMAD- instead of multiple starting points. Because the problem is especially expensive, NOMAD has been tested with only one seed, and not four, in order to reduce the number of optimization runs. The evaluation budget is $800(n + 1)$. As with the previous problem, only the best formulations from the aircraft range problem have been tested. The best one among them is SP8 for both smooth and nonsmooth alternatives. They manage to yield better results than no search, however, quad search is clearly the best algorithm on this problem.

Figure 10 shows the data profiles of SP8 with smooth and nonsmooth alternatives along with quad search. DFN and no search are not represent because they do not manage to solve one problem even at the largest tolerance. At tolerance $\tau = 10^{-1}$, the smooth alternative manages to solve as many problems as quad search. However, for higher tolerances, the latter is by far the best alternative. The poor performance of the extended aggregate models can be attributed to the stochasticity of the problem that importantly deteriorates the models. Nonetheless, it also impacts quad search. The superior performance of the latter is also due to the nature of the constraints: three out of five are linear, and one is cubic, which gives a non negligible advantage to quadratic models.
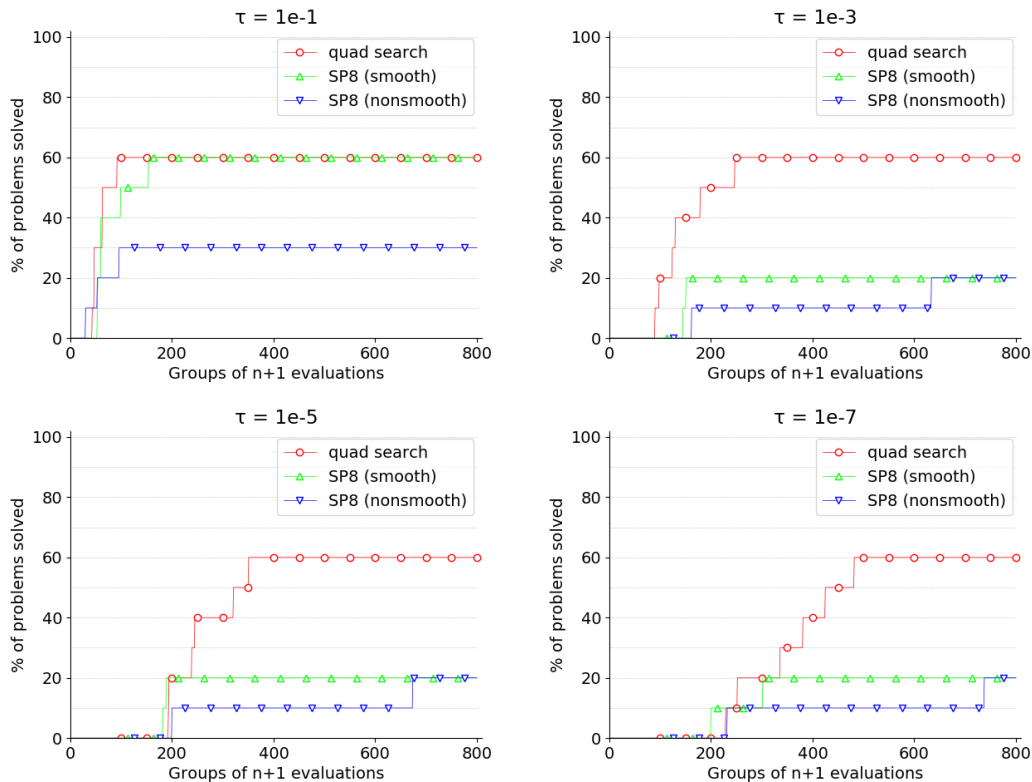

Figure 10: Data profiles. quad search vs. SP8 with smooth uncertainty vs. SP8 with nonsmooth uncertainty on solar1.

4.5   The styrene **problem**

The styrene problem is a chemical engineering simulator for styrene production described in [6] and available at github.com/bbopt/styrene. It aims at maximizing the net present value of the styrene

production process under structural, chemical and financial constraints with variables comprised of physical parameters and structure. The problem is deterministic but nonsmooth and with omnipresent hidden constraints, i.e., the simulation often fails to return a value even when all constraints are met. A random sampling resulted in almost 60% of failures in [30]. In addition, four constraints are binary. The problem has $n = 8$ bounded variables and $m = 11$ constraints.

Feasible regions may by especially hard to find on this problem. Consequently, twelve starting points were generated in three relatively easy regions. The evaluation budget is $600(n + 1)$. Like the previous problem solar1, the same subset of the most promising formulations found on the aircraft range problem has been tested. The best formulations are $SP1_1$ with the smooth alternative and $SP3_{0.01}$ with the nonsmooth alternative.

Figure 11 shows the data profiles of no search, quad search, $SP1_1$ with the smooth uncertainty and $SP3_{0.01}$ with the nonsmooth uncertainty at tolerance $\tau = 10^{-1}$ and $\tau = 10^{-2}$. On this problem, the extended aggregate models perform better than both no search and quad search. Besides, unlike all the others problems the latter yields worse results than no search due to the binary constraints. DFN struggles to find feasible solutions and the results are not presented.
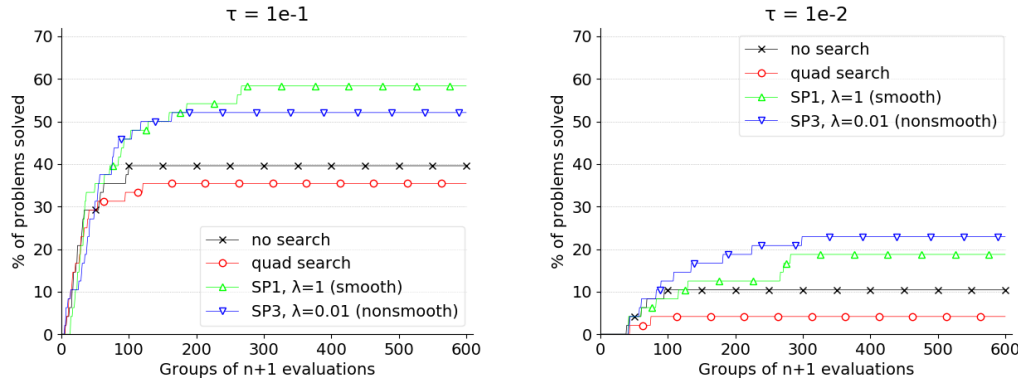

Figure 11: Data profiles. no search vs. quad search vs. $SP1_1$ with smooth uncertainty vs. $SP3_{0.01}$ with nonsmooth uncertainty on styrene.

4.6   **Results of** SHEBO

Unlike the other algorithms in this work, SHEBO does not take into account a starting point as an input, thereby making the comparison through data profiles impossible. In order to analyze the results, the best value found for a given optimization run is denoted by f^*, the real time (in minutes) needed to reach f^* is denoted by t^*, and the number of evaluations needed to reach f^* is denoted by k^*. Table 2 shows for most of the solvers seen before the median f^*, t^* and k^* on each problem considering all starting points and seeds, denoted by f_m^*, t_m^* and k_m^*, respectively, as well as the best f^* found on all runs, denoted by f_{best}^*, and the total real time needed for all runs, denoted by t_{tot}. For every problem, except the analytical problems, the three quantities are shown for no search, quad search, the best formulation found with the smooth uncertainty, the best formulation found with the nonsmooth uncertainty, DFN and SHEBO. However, SHEBO has been run only once on each problem due to its long running time. Consequently, all the values shown on the lines corresponding to SHEBO relate only to a single run, and f_m^* is equal to f_{best}^*.

For solar1 and styrene, SHEBO has a better f_{best}^* than the median f^* of all algorithms. However, on solar1 the proposed approach, both smooth and nonsmooth, manages to find a better f_{best}^* over all 10 runs after only 1.5 times more real time than the single run of SHEBO. On styrene, all the MADS algorithms manage to find a better f_{best}^* than SHEBO after all 48 runs, with significantly more real time though. On the aircraft range and the simplified wing problems, the proposed approach not only

Table 2: **Results of** SHEBO **compared to other algorithms.**

| | Aircraft range (40 runs except for SHEBO) | | | | | Simplified wing (40 runs except for SHEBO) | | | | |
	f_m^*	t_m^*	k_m^*	f_{best}^*	t_{tot}	f_m^*	t_m^*	k_m^*	f_{best}^*	t_{tot}
no search	-3964.204696	1	4140	-3964.204700	118	-16.4059	3	2405	-16.6119	397
quad search	-3964.204698	5	3064	-3964.204701	834	-16.5808	12	1291	-16.6120	2181
smooth	-3964.204699	10	3218	-3964.204701	1836	-16.6063	4	2115	-16.6120	921
nonsmooth	-3964.204699	9	2555	-3964.204700	1614	-16.5924	2	836	-16.6120	393
DFN	-1749.256329	0.1	73	-3143.101404	18	-15.0774	3	1134	-16.6112	588
SHEBO	·	784	4404	-3723.074357	3210	·	64	2190	-16.5501	356

| | solar1 (10 runs except for SHEBO) | | | | | styrene (48 runs except for SHEBO) | | | | |
	f_m^*	t_m^*	k_m^*	f_{best}^*	t_{tot}	f_m^*	t_m^*	k_m^*	f_{best}^*	t_{tot}
no search	-508184.0	190	4064	-660888.3	2384	-29305150	12	1541	-33613200	3611
quad search	-723623.2	510	3795	-835124.4	6837	-29301200	41	1675	-33000800	9750
smooth	-677669.7	305	3688	-849192.5	6303	-32704300	49	1809	-33705600	19015
nonsmooth	-658837.2	224	3359	-805799.5	6190	-32235250	48	1647	-33697400	16086
DFN	-332582.8	68	1448	-391849.8	735	-22851550	0.3	38	-28517600	49
SHEBO	·	2865	6190	-815086.2	4475	·	741	5365	-32873400	774

finds a better f_{best}^* than SHEBO but also a better f_{med}^* after only a fraction of the time per run. On the simplified wing problem, DFN also finds a better f_{best}^* than SHEBO with 1.7 times more real time. Regarding the number of evaluations, SHEBO requires more function evaluations to reach its best value than the other algorithms, except on the simplified wing problem. Overall, on the present experiments, one single run with SHEBO guaranties a decent value, only it demands a much larger computation time which is better invested running the proposed approach multiple times.

One final remark can be made about the real time required by the extended aggregate models compared to the quadratic models. The proposed approach is not always longer than quad search, both in terms of median time and total time. The quadratic models indeed tend to result in very long runs when they find a good bassin of solutions. Conversely, they result in very short runs when then they do not manage to find a good solution, depending on the starting point.

5   Discussion

This work proposes an extension to ensembles of models that enables to compute an uncertainty at any given point. The resulting extended aggregate models behave like stochastic models, i.e., they produce at any given point x a prediction $\hat{f}(x)$ and also an uncertainty $\hat{\sigma}(x)$, thus enabling to use tools inspired by Bayesian optimization. The proposed extended aggregate models are incorporated into the search step of MADS where at each iteration a surrogate subproblem derived from Bayesian optimization is solved in order to come up with new candidate points. The proposed approach may be used in any direct search method based on the search-poll paradigm, or in any approach akin to efficient global optimization if adapted. Any ensemble of models can be used along with any weight attribution technique provided that at least two models have a strictly positive weight at any moment.

The resulting algorithm has been tested on seven analytical problems, two multidisciplinary optimization problems and two simulation problems. The results show that the proposed extended aggregate

models incorporated into MADS find better solutions than MADS without search step or with the help of quadratic models on three expensive problems out of four. They also find better solutions than the stochastic models that they replace while requiring much less computational time. It should be noted that the models used to build the aggregate models must remain moderately expensive to compute, otherwise the method might loose its advantage in terms of computation time. The proposed approach does not show an advantage over quadratic models on analytical problems. In addition, the latter yield better results on the solar1 problem which most of the constraints are linear. The comparison to other solvers shows that the proposed approach has a clear advantage over DFN, and is more interesting than SHEBO when given the same computation time. An extended study of the various sub-problem formulations has not been conducted but based on the present results the formulations SP1, SP3 and SP8 can be recommended.

Future work may explore the uncertainty for the constraint independently from that of the objective, e.g., smooth uncertainty for the objective together with nonsmooth uncertainty for the constraint, since in this work the two were coupled. Other weight attribution techniques than that described in Section 3.2 may also be considered. The influence of the simplex used to build simplex gradients in (2) as well as the positive spanning set in (3) have not been studied in this work. Besides, the parameter λ of the surrogate subproblems has been carefully selected for each formulation but remains constant over the optimization once determined. It might instead be dynamically updated depending on the result of the search or merely follow a predetermined trend like decreasing with the number of iterations. The parameter α in (7) is proportional to the global variance of the cache. It could be refined in order to represent local trends better, for instance by taking into account the values of the cache only in a restricted area around the evaluated point, or by removing outliers. Finally, the formulations were chosen on a purely empirical basis. Little effort has been made to finely understand the behaviour and the performance thereof. A thorough analysis of the benefits of each formulations in the context of extended aggregate models might be a judicious undertaking.

A   Positive spanning set and simplex construction

The simplex and the positive spanning set described below are built in the *scaled* search space. Before constructing the models, the NOMAD software used in this work scales each input variable x_i, $i \in \{1, 2, \ldots, n\}$, using the mean and the variance of the points of the cache. This is done to give the same importance to all the variables regardless of their initial amplitude. Consequently, the simplex and the positive spanning set are isotropic in the scaled search space, but not in the actual search space.

- The simplex centred on $x \in \mathbb{R}^n$ used to build the simplex gradients $\nabla_S f(x)$ in Equation (2) is

$$\{x + 0.001 d_i \ : \ i \in \{1, 2, \ldots, n+1\}\}$$

where the directions d_i, $i \in \{1, 2, \ldots, n+1\}$, are constructed according to the following procedure:

$$d_i = \begin{cases} e_i - \dfrac{1 + \frac{1}{\sqrt{n+1}}}{n} \times [1, 1, \ldots, 1]^\top, & \text{if } i \in \{1, 2, \ldots, n\} \\ \dfrac{1}{\sqrt{2(n+1)}} \times [1, 1, \ldots, 1]^\top, & \text{if } i = n+1 \end{cases}$$

where e_i is the vector $[0, \ldots, 1, \ldots, 0]^\top$ with value 1 at the ith position. This procedure forms a regular simplex centred on x with side length equal to $\sqrt{2}$ in any dimension. The factor 0.001 was chosen empirically on preliminary tests.

- The positive spanning set centred on $x \in \mathbb{R}^n$ used in Equation (3) is

$$\{x \pm 0.005 e_i \ : \ i \in \{1, 2, \ldots, n\}\}$$

This positive spanning set contains $2n$ elements. The factor 0.005 was chosen empirically on preliminary tests.

B   Surrogate subproblem formulations

$$\min_{x \in \mathcal{X}} \ \hat{f}(x) - \lambda \hat{\sigma}_f(x) \tag{SP1-Fσ}$$
$$\text{s.t.} \ \ \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \ \ j = 1, 2, \ldots, m$$

$$\min_{x \in \mathcal{X}} \ \hat{f}(x) - \lambda \hat{\sigma}_f(x) \tag{SP2-FσP}$$
$$\text{s.t.} \ \ \text{P}(x) \geq p_c$$

$$\min_{x \in \mathcal{X}} \ - \text{EI}(x) - \lambda \hat{\sigma}_f(x) \tag{SP3-EIσ}$$
$$\text{s.t.} \ \ \hat{c}_j(x) - \lambda \hat{\sigma}_j(x) \leq 0, \ \ j = 1, 2, \ldots, m$$

$$\min_{x \in \mathcal{X}} \ - \text{EFI}(x) \tag{SP4-EFI}$$

$$\min_{x \in \mathcal{X}} \ - \text{EFI}(x) - \lambda \hat{\sigma}_f(x) \tag{SP5-EFIσ}$$

$$\min_{x \in \mathcal{X}} \ - \text{EFI}(x) - \lambda \hat{\sigma}_f(x) \mu(x) \tag{SP6-EFIμ}$$

$$\min_{x \in \mathcal{X}} \ - \text{EFI}(x) - \lambda (\text{EI}(x)\mu(x) + \text{P}(x)\hat{\sigma}_f(x)) \tag{SP7-EFICμ}$$

$$\min_{x \in \mathcal{X}} \ - \text{PFI}(x) \tag{SP8-PFI}$$

References

[1] M.A. Abramson, C. Audet, G. Couture, J.E. Dennis, Jr., S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. The NOMAD project. Software available at https://www.gerad.ca/nomad, 2021.

[2] M.A. Abramson, C. Audet, J.E. Dennis, Jr., and S. Le Digabel. OrthoMADS: A Deterministic MADS Instance with Orthogonal Directions. SIAM Journal on Optimization, 20(2):948–966, 2009.

[3] E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. Structural and Multidisciplinary Optimization, 37(3):279–294, 2009.

[4] C. Audet. Convergence Results for Generalized Pattern Search Algorithms are Tight. Optimization and Engineering, 5(2):101–122, 2004.

[5] C. Audet, V. Béchard, and J. Chaouki. Spent potliner treatment process optimization using a MADS algorithm. Optimization and Engineering, 9(2):143–160, 2008.

[6] C. Audet, V. Béchard, and S. Le Digabel. Nonsmooth optimization through Mesh Adaptive Direct Search and Variable Neighborhood Search. Journal of Global Optimization, 41(2):299–318, 2008.

[7] C. Audet and J. Côté-Massicotte. Dynamic improvements of static surrogates in direct search optimization. Optimization Letters, 13(6):1433–1447, 2019.

[8] C. Audet and J.E. Dennis, Jr. Mesh Adaptive Direct Search Algorithms for Constrained Optimization. SIAM Journal on Optimization, 17(1):188–217, 2006.

[9] C. Audet and J.E. Dennis, Jr. A Progressive Barrier for Derivative-Free Nonlinear Programming. SIAM Journal on Optimization, 20(1):445–472, 2009.

[10] C. Audet, J.E. Dennis, Jr., and S. Le Digabel. Parallel Space Decomposition of the Mesh Adaptive Direct Search Algorithm. SIAM Journal on Optimization, 19(3):1150–1170, 2008.

[11] C. Audet and W. Hare. Derivative-Free and Blackbox Optimization. Springer Series in Operations Research and Financial Engineering. Springer, Cham, Switzerland, 2017.

[12] C. Audet and W. Hare. Model-Based Methods in Derivative-Free Nonsmooth Optimization, pages 655–691. Springer International Publishing, 2020.

[13] C. Audet, M. Kokkolaras, S. Le Digabel, and B. Talgorn. Order-based error for managing ensembles of surrogates in mesh adaptive direct search. Journal of Global Optimization, 70(3):645–675, 2018.

[14] C. Audet, S. Le Digabel, V. Rochon Montplaisir, and C. Tribes. NOMAD version 4: Nonlinear optimization with the MADS algorithm. Technical Report G–2021–23, Les cahiers du GERAD, 2021.

[15] C. Audet and D. Orban. Finding optimal algorithmic parameters using derivative-free optimization. SIAM Journal on Optimization, 17(3):642–664, 2006.

[16] M. Björkman and K. Holmström. Global optimization of costly nonconvex functions using radial basis functions. Optimization and Engineering, 1:373–397, 2000.

[17] A.J. Booker. Well-conditioned Kriging models for optimization of computer simulations. Technical Report MandCT-TECH-00-002, Boeing Computer Services, Research and Technology, M/S 7L–68, Seattle, Washington 98124, 2000.

[18] A.J. Booker, J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, and M.W. Trosset. A Rigorous Framework for Optimization of Expensive Functions by Surrogates. Structural and Multidisciplinary Optimization, 17(1):1–13, 1999.

[19] L. Chen, H. Qiu, C. Jiang, X. Cai, and L. Gao. Ensemble of surrogates with hybrid method using global and local measures for engineering design. Structural and Multidisciplinary Optimization, 57(4):1711–1729, 2018.

[20] T.D. Choi, O.J. Eslinger, C.T. Kelley, J.W. David, and M. Etheridge. Optimization of automotive valve train components with implicit filtering. Optimization and Engineering, 1(1):9–27, 2000.

[21] F.H. Clarke. Optimization and Nonsmooth Analysis. John Wiley and Sons, New York, 1983. Reissued in 1990 by SIAM Publications, Philadelphia, as Vol. 5 in the series Classics in Applied Mathematics.

[22] A.R. Conn and S. Le Digabel. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. Optimization Methods and Software, 28(1):139–158, 2013.

[23] A.R. Conn, K. Scheinberg, and L.N. Vicente. Introduction to Derivative-Free Optimization. MOS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[24] A.R. Conn and Ph.L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In G. Di Pillo and F. Gianessi, editors, Nonlinear Optimization and Applications, pages 27–47. Plenum Publishing, New York, 1996.

[25] A.L. Custódio, H. Rocha, and L.N. Vicente. Incorporating minimum Frobenius norm models in direct search. Computational Optimization and Applications, 46(2):265–278, 2010.

[26] C. Davis. Theory of positive linear dependence. American Journal of Mathematics, 76:733–746, 1954.

[27] G. Fasano, G. Liuzzi, S. Lucidi, and F. Rinaldi. A Linesearch-Based Derivative-Free Approach for Nonsmooth Constrained Optimization. SIAM Journal on Optimization, 24(3):959–992, 2014.

[28] M. Lemyre Garneau. Modelling of a solar thermal power plant for benchmarking blackbox optimization solvers. Master's thesis, Polytechnique Montréal, 2015. Available at https://publications.polymtl.ca/1996/.

[29] T. Goel, R.T. Haftka, W. Shyy, and N.V. Queipo. Ensemble of surrogates. Structural and Multidisciplinary Optimization, 33(3):199–216, 2007.

[30] R.B. Gramacy and S. Le Digabel. The mesh adaptive direct search algorithm with treed Gaussian process surrogates. Pacific Journal of Optimization, 11(3):419–447, 2015.

[31] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh. Bayesian Optimization for Adaptive Experimental Design: A Review. IEEE Access, 8:13937–13948, 2020.

[32] W. Hock and K. Schittkowski. Test Examples for Nonlinear Programming Codes, volume 187 of Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, Germany, 1981.

[33] D.R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. Journal of Global Optimization, 21:345–383, 2001.

[34] D.R Jones, M. Schonlau, and W.J. Welch. Efficient Global Optimization of Expensive Black Box Functions. Journal of Global Optimization, 13(4):455–492, 1998.

[35] C.T. Kelley. Detection and remediation of stagnation in the Nelder-Mead algorithm using a sufficient decrease condition. SIAM Journal on Optimization, 10:43–55, 1999.

[36] S. Kitayama, M. Arakawa, and K. Yamazaki. Sequential approximate optimization using radial basis function network for engineering optimization. Optimization and Engineering, 12(4):535–557, 2011.

[37] S. Kodiyalam. Multidisciplinary aerospace systems optimization. Technical Report NASA/CR-2001-211053, Lockheed Martin Space Systems Company, Computational AeroSciences Project, Sunnyvale, CA, 2001.

[38] J. Larson, M. Menickelly, and S.M. Wild. Derivative-free optimization methods. Acta Numerica, 28:287–404, 2019.

[39] S. Le Digabel and S.M. Wild. A Taxonomy of Constraints in Simulation-Based Optimization. Technical Report G–2015–57, Les cahiers du GERAD, 2015.

[40] L. Lukšan and J. Vlček. Test problems for nonsmooth unconstrained and linearly constrained optimization. Technical Report V-798, ICS AS CR, 2000.

[41] A.L. Marsden, M. Wang, J.E. Dennis, Jr., and P. Moin. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. Journal of Fluid Mechanics, 572:13–36, 2007.

[42] M.D. McKay, R.J. Beckman, and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21(2):239–245, 1979.

[43] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum, pages 117–129. North-Holand, 1978.

[44] J.J. Moré and S.M. Wild. Benchmarking Derivative-Free Optimization Algorithms. SIAM Journal on Optimization, 20(1):172–191, 2009.

[45] J. Müller. An algorithmic framework for the optimization of computationally expensive bi-fidelity black-box problems. INFOR: Information Systems and Operational Research, 58(2):264–289, 2020.

[46] J. Müller and M. Day. Surrogate Optimization of Computationally Expensive Black-Box Problems with Hidden Constraints. INFORMS Journal on Computing, 31(4):689–702, 2019.

[47] J. Müller, J. Park, R. Sahu, C. Varadharajan, B. Arora, B. Faybishenko, and D. Agarwal. Surrogate optimization of deep neural networks for groundwater predictions. Journal of Global Optimization, 2020.

[48] J. Müller and R. Piché. Mixture surrogate models based on Dempster-Shafer theory for global optimization problems. Journal of Global Optimization, 51(1):79–104, 2011.

[49] J.A. Nelder and R.W.M. Wedderburn. Generalized Linear Models. Journal of the Royal Statistical Society: Series A (General), 135:370–384, 1972.

[50] N. Peremezhney, E. Hines, A. Lapkin, and C. Connaughton. Combining gaussian processes, mutual information and a genetic algorithm for multi-target optimization of expensive-to-evaluate functions. Engineering Optimization, 46(11):1593–1607, 2014.

[51] C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning. The MIT Press, 2006.

[52] R.G. Regis. Stochastic Radial Basis Function Algorithms for Large-scale Optimization Involving Expensive Black-box Objective and Constraint Functions. Computers and Operations Research, 38(5):837–853, 2011.

[53] R.G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. Engineering Optimization, 46(2):218–243, 2014.

[54] R.G. Regis and C.A. Shoemaker. Constrained global optimization of expensive black box functions using radial basis functions. Journal of Global Optimization, 31:153–171, 2005.

[55] X. Ruan, P. Jiang, Q. Zhou, J. Hu, and L. Shu. Variable-fidelity probability of improvement method for efficient global optimization of expensive black-box problems. Structural and Multidisciplinary Optimization, 62:3021–3052, 2020.

[56] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. Statistical Science, 4(4):409–435, 1989.

[57] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, and N. De Freitas. Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE, 104(1):148–175, 2015.

[58] N. Srinivas, A. Krause, S.M. Kakade, and M.W. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. Technical Report 0912.3995, arXiv, 2010.

[59] M.A. Taddy, R.B. Gramacy, and N.G. Polson. Dynamic trees for learning and design. Journal of the American Statistical Association, 106(493):109–123, 2011.

[60] B. Talgorn, C. Audet, M. Kokkolaras, and S. Le Digabel. Locally weighted regression models for surrogate-assisted design optimization. Optimization and Engineering, 19(1):213–238, 2018.

[61] B. Talgorn, S. Le Digabel, and M. Kokkolaras. Statistical Surrogate Formulations for Simulation-Based Design Optimization. Journal of Mechanical Design, 137(2):021405–1–021405–18, 2015.

[62] H.A. Le Thi, A.I.F. Vaz, and L.N. Vicente. Optimizing radial basis functions by d.c. programming and its use in direct search for global derivative-free optimization. TOP, 20(1):190–214, 2012.

[63] D.J.J. Toal. Some considerations regarding the use of multi-fidelity Kriging in the construction of surrogate models. Structural and Multidisciplinary Optimization, 51(6):1223–1245, 2015.

[64] V. Torczon. On the convergence of pattern search algorithms. SIAM Journal on Optimization, 7(1):1–25, 1997.

[65] R. Tournemenne, J.-F. Petiot, B. Talgorn, M. Kokkolaras, and J. Gilbert. Brass Instruments Design Using Physics-Based Sound Simulation Models and Surrogate-Assisted Derivative-Free Optimization. Journal of Mechanical Design, 139(4):041401–01–041401–9, 2017.

[66] C. Tribes, J.-F. Dubé, and J.-Y. Trépanier. Decomposition of multidisciplinary optimization problems: formulations and application to a simplified wing design. Engineering Optimization, 37(8):775–796, 2005.

[67] A. Verdério and E. W. Karas. On the construction of quadratic models for derivative-free trust-region algorithms. EURO Journal on Computational Optimization, 5(4):501–527, 2017.

[68] F.A.C. Viana, R.T. Haftka, S. Valder, Jr., S. Butkewitsch, and M.F. Leal. Ensemble of Surrogates: a Framework based on Minimization of the Mean Integrated Square Error. In 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials, Schaumburg, IL, 2008.

[69] F.A.C. Viana, R.T. Haftka, and L.T. Watson. Efficient global optimization algorithm assisted by multiple surrogate techniques. Journal of Global Optimization, 56(2):669–689, 2013.

[70] K.K. Vu, C. D'Ambrosio, Y. Hamadi, and L. Liberti. Surrogate-based methods for black-box optimization. International Transactions in Operational Research, 24(3):393–424, 2017.

[71] S.M. Wild, R.G. Regis, and C.A. Shoemaker. ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions. SIAM Journal on Scientific Computing, 30(6):3197–3219, 2008.

[72] S.M. Wild and C.A. Shoemaker. Global convergence of radial basis function trust region derivative-free algorithms. SIAM J. Optimization, 21(3):761–781, 2011.

[73] P. Ye, G. Pan, and Z. Dong. Ensemble of surrogate based global optimization methods using hierarchical design space reduction. Structural and Multidisciplinary Optimization, 58:537–554, 2018.